Customized Agents with Initiative

CS224V Project Pitch
In HW1-3 you trained agents that use the transaction state machine.
Transaction SM is very flexible - but not enough.
Agent is completely reactive.
Can we write an agent that has some initiative?
- Ask questions
- Make recommendations
- Offer follow-ups
- Help the user navigate content
function policy() {
    say("Hello! What is your name?");
    expect("{My name is|I am} ${name}\", { 
        name: Templates.constant_String
    }, (state, name) => makeSimpleState(state, 'set_name', name));
    const cmd = get({ acceptActs: ['set_name'] });
    const name = getName(cmd);
    say("Nice to meet you ${name}\", { name });
}

<table>
<thead>
<tr>
<th>Policy function</th>
<th>Say: what the agent says</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expect: what we expect the user to say</td>
<td></td>
</tr>
<tr>
<td>Get: what the user actually says</td>
<td></td>
</tr>
</tbody>
</table>
THE UGLY TRUTH: GENIE DOESN'T LOOK LIKE THAT (YET)

- Pseudo-code presented is a new, programmatic way to design agents
- (Internal code name: Dialogue PL)
- Currently an experimental branch: wip/dialogue-pl
- Inference (runtime) works well enough
- Synthesis is incomplete and a bit buggy
WHAT CAN YOU DO WITH DIALOGUE PL SO FAR? (OPTION 1)

Hybrid approach:

• Train a model with a transaction state machine
• At runtime, run the agent written in Dialogue PL
• Model transfers because/as long as:
  ○ Transaction synthesis includes some agent initiative, so you can ask questions
  ○ Some basic commands are handled by rules
  ○ Non-contextual commands would be handled by the model anyway
    ■ In the limit, you can track state by rules
EXAMPLE: NEWS NAVIGATION

In transaction SM:

U: Tell me the news

A: Here are the news

A: <plays headlines, one by one>
EXAMPLE: NEWS NAVIGATION

U: Tell me the news

A: Here is the top headline in the New York Times: "Small Needles, Short Lines: Biden’s Plan to Vaccinate Young Children". Would you like to hear it?

U: Yes

A: <play summary> . Would you like to hear the whole article?

U: No

A: Are you interested in more news about COVID-19, Joe Biden, or Children?

U: Tell me about Biden

A: Here is an article about Joe Biden in the New York Times: "Biden Heads to Scranton to Sell a Shrinking Agenda". Would you like to hear it?
What can you do with dialogue pl? (Option 2)

Forego Transactions:

- Write a new agent that doesn't use transactions at all (like the Hello World example)
- Basic commands (yes/no and multiple choice questions) always available without training
- Synthesis might work for simple dialogues
- Good enough for a dialogue tree, probably
GENERAL REMARKS

- This is not a project for the faint-hearted
- There will be bugs

But:

- You can design the agent however you like
- You can push the limit of what agents can do (mixed initiative)
- I will help you fix the bugs
PRACTICAL NOTES (IF YOU'RE STILL INTERESTED)

- The branch is wip/dialogue-pl in [genie-toolkit](https://github.com/stanford-oval/genie-toolkit)
- I'll keep working on that branch, so fork+branch off that
- After you switch branch, do `git clean -fdx lib/` followed by `npx make` otherwise the build will fail

- Starter code from HW3 should be adequate for this project
- Use `npm link` to link genie-toolkit from the branch into the starter code
- It should work on Mac/Windows too (if you have the deps), you don't have to be in GCP

- At this time, you have to edit that file, there is no way to load a different one
- The `get` and `say` function are in the `DialogueInterface` class
AVAILABLE DOCUMENTATION

- You can build API docs of genie-toolkit with `npm run doc`, the api docs will be in jsdoc/
- Examples of dialogue PL are https://wiki.genie.stanford.edu/genie/design-notes/dialogue-pl/examples
BUILDING A WORLD-WIDE
VOICE WEB

CS224V Project Pitch
BACKGROUND AND MOTIVATION

- Thingpedia (at its peak) had ~150 skills
- Mycroft has ~50 skills
- Alexa has 250k skills
- How can open-source agents compete with proprietary platforms?

- But actually, is 250k really good?
- 1.2B websites
- Idea: leverage the open web
The World Wide VOICE Web architecture

- No more skill repositories
  - No more walled gardens
- Every website can host a voice agent directly
  - Offer it on the website itself
  - On messaging, phone, mobile apps
- Smart speakers can discover the voice agents using standard web tech
  - Web protocols
  - Search engines
PRETRAINED AGENTS: TRAIN ONCE, DEPLOY ANYWHERE

- Pretrained agents are agents that use abstract APIs
- Can be built once and for all for a domain
- Any website can adopt them by implementing the abstract APIs

- In ThingTalk: abstract classes
- Examples: IoT, media players
THIS PROJECT: LET'S BUILD THE DISCOVERY MECHANISM

- How does the smart speaker map a name to a website?
  - The users won't say "h t t p colon slash slash n y times dot com"
  - Or worse "h t t p colon slash slash slash dot dot com"
- How does the smart speaker learn about an agent in the website?
- How are APIs exposed?
- What can kind of agents can we support?
1. Users says "Launch New York Times"
2. Use Google/Bing to map "New York Times" to nytimes.com
3. Load well-known URL containing a ThingTalk manifest
4. ThingTalk manifest contains either:
   a. Pretrained agent reference + URL endpoints for APIs
   b. Completely custom agent + URL endpoints for NLP (and speech?)
CHALLENGES, OPEN QUESTIONS

- Is there an index/directory of WWvW agents?
- What does a WWvW manifest look like?
- For a pretrained agent, does the agent need to load code from the website?
  - If so, how is it safe?
  - If not, is there a standard RPC protocol that the website offers?
- How does the agent interact with custom agents?
  - Is it text, speech, ThingTalk?
  - Can the custom agent control the client HW (playback, leds, etc.)?
- How does the assistant switch to and from custom agents?
  - Are there "builtin" commands to enter and leave?
  - Can you enter a custom agent with a pretrained agent command?
  - Can data flow from one custom agent to the next?
PRACTICAL NOTES

- Familiarize yourself with ThingTalk manifest format
- Look at the existing skills at https://github.com/stanford-oval/thingpedia-common-device
- Code to load skills is at https://github.com/stanford-oval/thingpedia-api